

Unidad I: Introducción a C#

Lic. Ronaldo Armando Canizales Turcios

Departamento de Electrónica e Informática, UCA

Ciclo virtual 01-2021

Agenda

1 Sentencias de control de flujo

2 Funciones

3 Introducción a Listas

4 Anuncios

1. Sentencias de control de flujo

Sintaxis del for

```
for (int i = 0; i < 10; i++) {  
    System.out.println(i * 2);  
}
```

Sintaxis del while

```
int i = 0;  
while (i < 10) {  
    System.out.println(i);  
    i++;  
}
```

Sintaxis del do-while

```
int k = 0;  
do {  
    System.out.println(k);  
    k++;  
} while(k < 10);
```

Ejemplo: Cálculo de CUM de ciclo

Las unidades de mérito de una asignatura cursada se obtienen al multiplicar la calificación final de la asignatura por sus unidades valorativas.

El CUM se obtiene al dividir el total de unidades de mérito obtenidas entre el total de las unidades valorativas de las asignaturas cursadas.

	Nota obtenida	Unidades Valorativas (U.V.)	Unidades de mérito (nota obtenida multiplicada por U.V.)
CICLO I			
Fundamentos de Administración	8.2	3	24.6
Introducción a la Economía I	7.4	4	29.6
Principios de Contabilidad	5.8	4	23.2
Principios de Matemática	6.6	4	26.4
SUMATORIA		15	103.8



Para calcular CUM del ciclo dividir 103.8 entre 15

CUM calculado al final del Ciclo I 6.92

1. Sentencias de control de flujo

La sintaxis varía un poco en C# respecto a C++

- La sintaxis para la creación de arreglos:
 - `char[] vocales1 = new char[5];`
 - `string[] vocales2 = new string[5]{ "a", "e", "i", "o", "u"};`
 - `string[] vocales3 = new string[]{ "a", "e", "i", "o", "u"};`
- **foreach**: recorrer un arreglo o una lista.

Ejemplo: crear arreglo de nombres

Solicitar al usuario una lista de nombres separados por coma, almacenar cada uno de ellos individualmente en una casilla de un arreglo.

Finalmente mostrar cada nombre por separado.



2. Funciones

En lenguajes de programación orientados a objetos, como C#, no se acostumbra utilizar el paradigma estructurado, aunque sí es posible hacerlo.

```
public static void saludar() {  
    System.out.println("Hola Mundo!");  
}
```

Es posible ejecutar funciones desde el *main* sólo si se les antecede la palabra reservada **static** (la explicación formal de esto se dará en clases posteriores).

2. Funciones

A diferencia de C++, en C# no se utilizan los prototipos de funciones. Las funciones pueden declararse en cualquier orden, antes o después del *main*.

```
public static void main(String[ ] args) {  
    String result = isLeap("09/03/2020") ? "Si" : "No";  
    System.out.println(result + " es bisiesto.");  
}  
public static boolean isLeap(String date) {  
    String s_year = date.Substring(6);  
    int i_year = Convert.ToInt32(s_year);  
    return (i_year % 4 == 0);  
}
```

2. Funciones

Ejemplo: Habitaciones de un hotel

Implemente un programa que almacene la cantidad de habitaciones ocupadas actualmente y las despachadas de manera histórica para un pequeño hotel que tiene diez cuartos, cada uno vale \$15.

Se deberá mostrar el siguiente menú:

- 1) Reservar habitaciones.
- 2) Despachar habitaciones.
- 3) Habitaciones ocupadas.
- 4) Calcular ganancias.
- 5) Salir.



Finalmente, genere un EXE para poder ejecutar su programa.

2. Funciones

Sobrecarga

Se refiere a la posibilidad de tener dos o más funciones con el mismo nombre pero funcionalidad diferente. El compilador usará una u otra dependiendo de los parámetros usados. Las funciones deben contar con diferentes argumentos, caso contrario se generará un error.

```
public static int suma(int x, int y) {  
    return x + y;  
}  
public static double suma(double a, double b) {  
    return a + b;  
}  
public static double suma(double a, double b, double c) {  
    return a + b + c;  
}
```

3. Introducción a Listas

Es bastante similar (inclusive a las librerías que manejamos en C++) pero en C# contamos con la poderosa librería Linq, la iremos viendo poco a poco con el tiempo, ya que será de gran utilidad. Por ejemplo:

```
List<int> lista = new List<int>();  
lista.Add(1);  
lista.Add(2);  
lista.Insert(1, 5);  
  
foreach (int numero in lista)  
    Console.Write(numero + " ");  
  
Console.WriteLine(lista.Max());  
Console.WriteLine(lista.Sum());  
lista.RemoveAt(1);
```



3. Introducción a Listas

Ejemplo: Conversor de temperaturas

Implemente un programa que almacene una cantidad dinámica de temperaturas, originalmente en grados Celsius. El usuario podrá agregar o eliminar datos en cualquier momento. Se deberá mostrar el siguiente menú:

- 1) Agregar temperatura (Celsius).
- 2) Quitar temperatura (Celsius).
- 3) Mostrar temperatura máxima (Celsius).
- 4) Mostrar temperaturas en Celsius.
- 5) Mostrar temperaturas en Fahrenheit.
- 6) Mostrar temperaturas en Kelvin.
- 7) Salir.

$$^{\circ}\text{F} = (^{\circ}\text{C} \times \frac{9}{5}) + 32$$

$$^{\circ}\text{C} = (^{\circ}\text{F} - 32) \times \frac{5}{9}$$

$$^{\circ}\text{C} = (\text{K} - 273.15)$$

$$\text{K} = (^{\circ}\text{C} + 273.15)$$

4. Anuncios

Charla informativa IEEE

Fecha
Viernes 12 de Marzo

Horarios
11:00 am - 11:30 am
4:30 pm - 5:00 pm



Contactos

 uca.ieeesv@gmail.com
 ibarra.veronica@ieee.org

 ieeeucasv

 IEEE-UCA



Universidad Centroamericana
José Simeón Cañas
IEEE Student Branch

Como Rama Estudiantil IEEE-UCA te invitamos a nuestra charla informativa donde conocerás:

- ¿Qué es IEEE?
- Beneficios de la membresía.
- Áreas de interés de nuestros capítulos técnicos y grupos de afinidad.
- Diagrama de actividades que se realizarán a lo largo del año:
 - Talleres y Cursos.
 - Charlas técnicas.
 - Charlas no técnicas.
 - Eventos sociales.
 - Competencias.
- ¿Cómo participar en nuestros proyectos para obtener horas sociales?

A través de



¡Te esperamos!



IEEE SIGHT
Special Interest Group on Humanitarian Technology



IEEE COMPUTER SOCIETY



IEEE PES
Power & Energy Society



IEEE IAS
IEEE Industry Applications Society



IEEE ComSoc
IEEE Communications Society



IEEE Women in Engineering

Unidad I: Introducción a C#

Lic. Ronaldo Armando Canizales Turcios

Departamento de Electrónica e Informática, UCA

Ciclo virtual 01-2021